

In the Claims:

Please amend Claims 1-3, 7, 15-17, 21, 30-32, and 36, add Claim 44, and cancel Claim 29, all as shown below. Applicant respectfully reserves the right to prosecute any originally presented claims in a continuing or future application.

1. (Currently Amended) A method for dynamically binding a user interface to information, comprising:

presenting a user interface to a user wherein the user interface is operable to present information stored in a business object to the user, collect user information from the user, and store the user information in the business object;

defining a data binding tag wherein the data binding tag includes a plurality of attributes;

specifying a first action by the data binding tag;

specifying, using a scripting language, at least one attribute on the data binding tag to reference the first data source associated with the ~~with a first language~~ a first action, wherein the first data source is in the business object;

~~specifying with a second language a first data source associated with the first action;~~

rendering each item in the first data source in the user interface output ~~with a third markup language based at least partially on the first action; and~~

~~wherein the second language is embedded in the first language; and~~

wherein the first action can set or get the first data source.

2. (Currently Amended) The method of claim 1 wherein:

the ~~first language~~ data binding tag allows for the specification of JavaServer Page action elements.

3. (Currently Amended) The method of claim 1 wherein:

the ~~second~~ scripting language is based on the Javascript language.

4. (Original) The method of claim 1 wherein:

the first data source identifies one of: 1) an object field; 2) an object property; and 3) an Extensible Markup Language document element.

5. (Original) The method of claim 4 wherein:
an object is a JavaBean.
6. (Original) The method of claim 1 wherein:
the first data source is one of: 1) an array; 2) a list; 3) a map.
7. (Currently Amended) The method of claim 1 wherein:
the ~~third~~markup language can include at least one of: Hypertext Markup Language (HTML), Dynamic HTML, Extensible HTML, and Extensible Markup Language.
8. (Original) The method of claim 1 wherein:
the first action can be a child of another action.
9. (Original) The method of claim 1 wherein:
the first action can have at least one child action.
10. (Original) The method of claim 9 wherein:
the at least one child action can have at least one other child action.
11. (Original) The method of claim 9 wherein:
the at least one child action can selectively process the first data source.
12. (Original) The method of claim 9 wherein:
the at least one child action can refer to the first data source with a context defined by the first action.
13. (Original) The method of claim 9 wherein:
the at least one child action can perform at least one of the following actions on the first data source: 1) set; 2) get; 3) sort; and 4) filter.
14. (Original) The method of claim 9, further comprising:
rendering a list or a table based on the first data source.

15. (Currently Amended) A machine readable medium having instructions stored thereon that when executed by a processor cause a system to:

present a user interface to a user wherein the user interface is operable to present information stored in a business object to the user, collect user information from the user, and store the user information in the business object;

~~specify~~ define a data binding tag, wherein the data binding tag specifies with a first language a first action and wherein the data binding tag includes a plurality of attributes;

specify, using a scripting language, with at least one attribute on the data binding tag to reference a second language a first data source associated with the first action, wherein the first data source is in the business object;

render each item in the first data source in the user interface output with a third markup language based at least partially on the first action; and

~~wherein the second language is embedded in the first language; and~~

~~wherein the first action can set or get the first data source.~~

16. (Currently Amended) The machine readable medium of claim 15 wherein:

~~the first language~~ data binding tag allows for the specification of JavaServer Page action elements.

17. (Currently Amended) The machine readable medium of claim 15 wherein:

~~the second~~ scripting language is based on the Javascript language.

18. (Original) The machine readable medium of claim 15 wherein:

the first data source identifies one of: 1) an object field; 2) an object property; and 3) an Extensible Markup Language document element.

19. (Original) The machine readable medium of claim 18 wherein:

an object is a JavaBean.

20. (Original) The machine readable medium of claim 15 wherein:

the first data source is one of: 1) an array; 2) a list; 3) a map.

21. (Currently Amended) The machine readable medium of claim 15 wherein:
the ~~third~~ markup language can include at least one of: Hypertext Markup Language (HTML), Dynamic HTML, Extensible HTML, and Extensible Markup Language.
22. (Original) The machine readable medium of claim 15 wherein:
the first action can be a child of another action.
23. (Original) The machine readable medium of claim 15 wherein:
the first action can have at least one child action.
24. (Original) The machine readable medium of claim 23 wherein:
the at least one child action can have at least one other child action.
25. (Original) The machine readable medium of claim 23 wherein:
the at least one child action can selectively process the first data source.
26. (Original) The machine readable medium of claim 23 wherein:
the at least one child action can refer to the first data source with a context defined by the first action.
27. (Original) The machine readable medium of claim 23 wherein:
the at least one child action can perform at least one of the following actions on the first data source: 1) set; 2) get; 3) sort; and 4) filter.
28. (Original) The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:
render a list or a table based on the first data source.
29. (Canceled)

30. (Currently Amended) A software framework for rendering at least one object on a user interface, comprising:

~~a first language capable of specifying a first action~~ set of data binding tags specified in a JSP programming language, wherein the data binding tags include a plurality of attributes to reference and display data, and wherein each tag can be used to bind and submit data that a user may edit in a web page;

wherein each attribute can include an expression, written in an expression language, that ~~a second language capable of specifying references~~ a first data source business object that includes data collected from or presented to the user ~~associated with the first action; and~~

~~a third markup language capable of rendering each item in the first business object in a user interface that is referenced by the expression, wherein the expression is evaluated at rendering.~~ output based at least partially on the first action;

~~wherein the second language is embedded in the first language; and~~

~~wherein the first action can set or get the first data source.~~

31. (Currently Amended) The framework of claim 30 wherein:

~~the first language~~ data binding tag allows for the specification of JavaServer Page action elements.

32. (Currently Amended) The framework of claim 30 wherein:

~~the second~~ scripting language is based on the Javascript language.

33. (Original) The framework of claim 30 wherein:

the first data source identifies one of: 4) an object field; 2) an object property; and 3) an Extensible Markup Language document element.

34. (Original) The framework of claim 33 wherein:

an object is a JavaBean.

35. (Original) The framework of claim 30 wherein:

the first data source is one of: 4) an array; 2) a list; 3) a map.

36. (Currently Amended) The framework of claim 30 wherein:
the ~~third~~ markup language can include at least one of: Hypertext Markup Language (HTML), Dynamic HTML, Extensible HTML, and Extensible Markup Language.
37. (Original) The framework of claim 30 wherein:
the first action can be a child of another action.
38. (Original) The framework of claim 30 wherein:
the first action can have at least one child action.
39. (Original) The framework of claim 38 wherein:
the at least one child action can have at least one other child action.
40. (Original) The framework of claim 38 wherein:
the at least one child action can selectively process the first data source.
41. (Original) The framework of claim 38 wherein:
the at least one child action can refer to the first data source with a context defined by the first action.
42. (Original) The framework of claim 38 wherein:
the at least one child action can perform at least one of the following actions on the first data source: 4) set; 2) get; 3) sort; and 4) filter.
43. (Original) The framework of claim 38 wherein:
a list or a table can be rendered based on the first data source.
44. (New) A system for dynamically binding a user interface to information, comprising:
a computer including a computer readable medium and processor operating thereon;
a user interface that is operable to
 present information stored in a business object to a user,
 collect user information from the user, and

store the user information within the business object;

a plurality of data binding tags written in a first programming language stored on the computer readable medium wherein each data binding tag

- includes a lifecycle associated therewith,
- includes a plurality of attributes, and
- specifies an action to be performed on the business object wherein the action includes setting or getting information from the business object including the user information;

an expression language that can be used to evaluate expressions on specified business objects, wherein each expression specifies a business object in which the expression is to be evaluated; and

a markup language that can be used to render the specified business objects in the user interface.